RICE UNIVERSITY
School of Engineering

COMP 621: Systems Software

Spring 2021

Professor: John Greiner
Contact Information: greiner@rice.edu, Duncan Hall 3093

TA: Shaquille Que
Contact Information: stq1@rice.edu

Course Materials

The course's lecture-oriented and assessment materials will be available via the Canvas web site. Supplemental materials are available on the web.

This course does not use a textbook. However, the following texts are good supplements:

- "Systems Performance" by Brendan Gregg
- "BPF Performance Tools" by Brendan Gregg
- Any C reference book

Hardware Requirements

You will need a Raspberry Pi. For details on what you need and how to set it up, please see the document "COMP 621 Raspberry Pi Setup" in the course Canvas site, under Course Resources.

Why a Raspberry Pi?

- Software convenience -- Lots of free tools are available in Linux.
- Consistency -- Each student will use the same type of platform.
- Isolation -- Each student will use their own computer, so its behavior and performance is not affected by other students.

- Control -- Each student will be in control of their own computer, able to run any program and install any software.
- Inexpensive -- It's a lot cheaper than specifying a particular laptop.

While you could use the RP as your only computer, it will be more convenient to have another computer, of any type, to (1) browse the web and course contents and (2) connect to the Raspberry Pi.

For a few tasks in the course, you will instead use Rice's CLEAR Linux servers. We will arrange for COMP 621 students to get accounts with your NetID as the user ID.

Software Requirements

All the course software will be available via free download, mostly as Linux packages or from Git Classroom. We will provide instructions on how to download these materials.

Course Overview

Modern computer systems are designed and implemented in a layered fashion, wherein each layer builds upon those beneath it, providing abstractions for processing, memory, and I/O that are progressively more abstracted from the hardware and easier to use than those of the underlying layers. While this layered architecture has made building systems easier, it has also made understanding their behavior and performance more difficult. Every layer from the managed run-time environments used by modern programming languages to the hypervisor play a role in processor scheduling, memory management, and I/O, making it oftentimes difficult to pinpoint which layer of the system is interacting poorly with a program. This class will teach students about the fundamental characteristics of the abstractions for processing, memory, and I/O at each layer of a modern computer system, so that they might understand the functionality provided by each layer, and instruct them on the use of modern debugging, profiling, and tracing tools, so that they are equipped to characterize the behavior and performance of their programs.

The course has three learning objectives, i.e., high-level goals:

- (Memory) Simulate memory hierarchy behavior
- (Concurrency) Write software to demonstrate different approaches to concurrency

- (Tools) Use modern debugging, profiling, and tracing tools to characterize the behavior and performance of programs

Prerequisites

The formal prerequisite is COMP 613, Programming Languages, so that you have sufficient programming experience in a variety of languages, and familiarity with Linux. More generally, the following describe the expectations of students.

- Data structures: Familiar with the concepts, but not necessarily the implementations, of arrays, stacks, queues, linked lists, trees, mappings.
- Algorithms: Familiar with the concepts, but not necessarily the implementations, of linear search, binary search, sorting in $O(n \log n)$ time. Can understand an algorithm presented in pseudo-code.
- Linux: Familiar with logging in, using commands like `ls` and `cd`, and using command arguments. We will introduce many more Linux commands, and you will use Linux extensively throughout the course.
- Programming: Can read simple code in C, plus a little Python and Java, with the help of more advanced features being explained to you. Later in the class, you will modify programs written in C. Can use and understand a programming reference manual.

Course Requirements

This course is still under development. The number of assignments, their content, and their due dates are tentative.

Activity	Deadline	Grade weight	Time estimate
Videos + in-video questions		0%	2 hrs/wk
Live “lab” session (13)	Th. night	0%	1 hr/wk
Problem set (13)	Sat. night	23%	1.5 hrs/wk
Feedback survey (13)	Sun. night	2%	0.25 hrs/wk
Assignments (8?)	Tue. night	40%	8 hrs each
Exams (2)		35%	3 hrs each

The time required for each activity, especially the assignments, can vary tremendously by student. The times listed in the above table are only estimates.

Your activities in this class are organized around the following schedule. However, the material is released early so that you will have lots of flexibility to do the activities prior to the deadlines.

Wednesday - Thursday: Each week will have a set of video lectures and a live hands-on lab session where you will work in groups. Some videos contain ungraded in-video questions. The Thursday night lab session will focus mostly on tool use and small programming tasks. Participation in the lab is strongly encouraged, but optional.

Saturday: There will be a problem set based upon the lecture and lab content. Each will consist of two parts, one computer-graded and one human-graded.

Sunday: Each week will also have a short anonymous feedback survey. You will provide feedback to the instructor and, in doing so, also reflect on the week's material.

Tuesday: Each major course topic will have one or two assignments. These will be due on Tuesday nights.

Other: There will also be a midterm exam at the half-way point, plus a final exam after the last week.

Attendance Policy

Attendance is not required, but strongly encouraged for all live class sessions. Your grade will not be penalized if you don't attend.

However, a primary benefit of this program is the time spent during these live sessions interacting with your classmates and faculty, with much of the learning coming from these interactions. These live sessions are a major part of not only learning the content, but also the experience as a whole. As stated above, one of their primary purposes is to clarify any misconceptions that you might have. Additional content examples and details beyond the video lectures will be in the live sessions.

RICE UNIVERSITY
School of Engineering

Contacting Staff

For non-personal questions, such as content and assignment clarifications, the best forum is Canvas Discussions. After all, if you have a question, it is likely that other students have the same question and would like to know the answer.

For personal questions about your assignment, grades, or such, the best way to contact me is via email, greiner@rice.edu. Please use a detailed subject line including "COMP 621", since I teach multiple courses.

Course Outline

The course will consist of six modules. Each module concludes with an assignment. The following summarizes the schedule and topics.

In accordance with this semester's policies, we will have a one-week break on the week of our "Sprinkle" day, April 8.

Week	Module	Topics	Assignment due
1	1.1	Big picture	
2	1.2	Observation	
3	1.3	Hardware	
4	2.1	Memory hierarchy & cache	1: 2/16/2021
5	2.2	Cache & virtual memory	
6	2.3	Memory allocation & deallocation	2a: 3/2/2021
7	3.1	I/O	2b: 3/9/2021
8	3.2	Processes & threads	
9	3.3	Concurrency issues	3a:3/23/2021
10	4.1	File systems	3b: 3/30/2021

11	4.2	Modern file systems	
12	5	Networking	4: 4/20/2021
13	6	Machine virtualization	5: 4/27/2021
			6: 5/4/2021 (incl. extension) Final: 5/13/2021

Assessment Deadlines and Grades

We publish course materials well before the corresponding deadlines so that students have the flexibility to work according to their own schedule. However, that flexibility ends at the deadline. Graded work will not be accepted after the posted deadline, unless there are exceptional and unforeseen circumstances.

Assignment Details

Assn.	Due Date	Points	Summary
1	2/16/2021	100	Short answer problems on background material
2a	3/2/2021	100	Write a cache simulator
2b	3/9/2021	100	Short answer problems on memory hierarchy
3a	3/23/2021	100	Write concurrent servers
3b	3/30/2021	100	Short answer problems on concurrency
4	4/20/2021	100	TCP performance analysis
5	4/27/2021	100	I/O latency performance analysis
6	5/2/2021 (incl. extension)	50	Virtual machine performance analysis Short answer problems on machine virtualization

RICE UNIVERSITY
School of Engineering

Canvas deadlines

Canvas has a “due by” and an “available until” deadline. The Canvas system will allow submissions until the “available until” deadline, but any submissions after the “due by” deadline will be marked late. As described, graded work will not be accepted after the posted deadline, so these two deadlines are the same.

Note that a posted deadline of 11:00pm, for example, is indeed zero seconds after 11:00pm. Canvas and the instructors will count an assignment late if it is turned in one or more seconds after the “due by” deadline, even though it may appear to be the same time. Also, the submission process takes some (small) amount of time, and your work is not submitted until the end of that process. So, be sure to **always** submit a few minutes before the deadline.

For most assessments, you will be allowed to make multiple submissions. Canvas allows you to look at your submission, and we **strongly recommend** that you check you successfully submitted what you intended. For quizzes, since they are automatically graded by computer, the **highest** score will be used. For exercises and assignments, the **last** submission will be graded.

Late submissions and extensions

As stated above, late submissions are not accepted. However, you may request an extension if there are exceptional and unforeseen circumstances. You do so by emailing the instructor with a justification and the designed extension length. Examples include emergency medical issues by immediate family or dependents, last-minute required work travel, and natural disasters. It is expected that students will have other classes, a job, families, or other responsibilities, and deadlines from such activities are **not** considered exceptional circumstances. You should request an extension as early as possible, as extension requests received after the deadline are received much less favorably.

The instructor has the flexibility to grant a class-wide extension in the event of unforeseen circumstances that affect a large portion of the class.

Grading

You will receive your assessment scores and feedback through the Canvas system. You can look either at the original assessment or go via the Canvas "Grades" feature.

You may request a regrading of your work if you think it was graded incorrectly. You do so by emailing the instructor with a specific description of which deductions you think were incorrect and why. Examples include that the grader overlooked or misinterpreted part of your answer. A regrade request must come within 7 days of your score and feedback being posted within Canvas.

Honor Code Policy

The Honor Code is a special privilege and responsibility at Rice University. You pledged to uphold the Rice Honor Code when you matriculated at Rice, and we take the Honor Code very seriously. The work you submit for this class is expected to be the result of your own work. Attempting to take credit for someone else's work by turning it in as your own constitutes plagiarism, as defined by the Rice Honor Code.

As stated, the guiding principle is that all graded work is to be done individually. However, you are encouraged to discuss problems with the course staff and classmates. The specific limits on access to materials varies by type of assessment.

All Masters of Computer Science students are sworn to adhere to the tenets set forth in Rice University's Honor Code. All written assignments for this class submitted to the instructor must carry the signed Honor Code pledge: "On my honor, I have neither given nor received any unauthorized aid on this (examination, quiz or paper)." (with signature and/or student ID number). Failure to include this pledge will result in an ungraded and returned assignment.

Please be aware of the Honor Code addendum that is part of the Student Handbook. This will provide more specific language about how the Honor Code should be adhered to as part of the Computer Science program.

All graded work other than exams

Allowed:

- Any material provided as part of the course, including videos, notes, and solutions.
- Your course work, including graded work and any notes.

- Any website, book, or other material. However, you may use at most the equivalent of two lines of code or math per assignment (or other work), and you must reinterpret any text in your own words. You must cite any source used.
- Communicating with course staff
- Communicating with other students in the class about general approaches to problems, but not about specific solutions.

Not allowed (an incomplete list):

- Copying large parts of solutions for any source other than what is provided as part of the course or your own work
- Copying small parts of solutions *without citation* from any source other than what is provided as part of the course or your own work.
- Communicating with other students in the class about a specific solution
- Communicating with anyone outside the class. This includes doing things like posting questions to StackOverflow.
- Posting sample solutions to any public forum.

Exams

Allowed:

- Any material provided as part of the course, including videos, notes, and solutions.
- Your course work, including graded work and any notes.
- The official reference material for any application used as part of the course. The course will provide a specific list.
- Communicating with course staff

Not allowed (an incomplete list):

- Any other website, book, or other material
- Communicating with any other person about the exam
- Posting sample solutions to any public forum.

Exams will use the ProctorU service to monitor students. This requires that students use a Windows or MacOS computer for the exam.

Disability-based Accommodations

If you have a documented disability that may affect academic performance, you should: 1) make sure this documentation is on file with Disability Resource Center (Allen Center, Room

111 / adarice@rice.edu / x5841) to determine the accommodations you need; and 2) meet with me to discuss your accommodation needs.

Title IX Policy

Rice University cares about your wellbeing and safety. Rice encourages any student who has experienced an incident of harassment, pregnancy discrimination, gender discrimination, or relationship, sexual, or other forms interpersonal violence to seek support through The SAFE Office. Students should be aware when seeking support on campus that most employees, including myself, as the instructor/TA, are required by Title IX to disclose all incidents of non-consensual interpersonal behaviors to Title IX professionals on campus who can act to support that student and meet their needs. For more information, please visit safe.rice.edu or email titleixsupport@rice.edu .